08-31-00

A

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

August 30, 2000

Honorable Assistant
Commissioner of Patents
Washington, D.C. 20231

SUBJECT: Patent Application
Inventors: Donald C. Englin; Kelvin S. Vartti; and James L. Federici
Title: Method for Managing Flushes With the Cache
File No: RA 5265 (33012/294/101)

Dear Sir:

Enclosed herewith are the following papers comprising an application for patent as identified above:

1. Specification (17 pages)
2. Claims (6 pages)
3. Formal Drawings (5 pages)
4. Declaration and Power of Attorney
5. Assignment of Invention
6. Assignment Coversheet

Please charge the Assignment fee of $40.00 and the Patent Application filing fee of $768.00, calculated below, to Account No. 19-3790 of Unisys Corporation. If the calculated fee is incorrect, you are authorized to charge the correct fee.

The filing fee was calculated as follows:

1.	Basic Fee	$690.00
2.	Additional Fees	
	a. Number of claims in excess of 20, (20-20=0) 0 times $18	0.00
	b. Number of independent claims minus 3, (4-3=1) 1 times $78	78.00

TOTAL 768.00

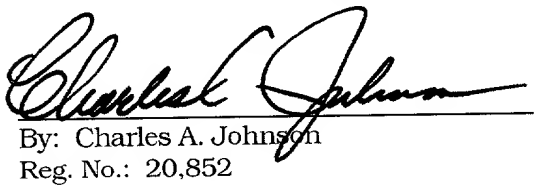
Correspondence is to be directed to the undersigned attorney of record, and an early acknowledgment will be greatly appreciated.

Respectfully submitted,

Charles A. Johnson
Reg. No. 20,852
Unisys Corporation
P O Box 64942 MS 4773
St. Paul, MN 55164
Attorney for Applicants

CAJ/rjn
(651)635-7702
Enclosures

CERTIFICATE UNDER 37 CFR 1.10: The undersigned hereby certifies that this transmittal letter and the paper of papers, as described hereinabove, are being deposited in the United States Postal Service, "Express Mail Post Office to Addressee" having an Express Mail mailing label number of EL 027383825 US, in an envelope addressed to: COMMISSIONER OF PATENTS AND TRADEMARKS, Washington, D.C. 20231 on this 30th day of August 2000.



By: Charles A. Johnson
Reg. No.: 20,852

APPLICATION FOR UNITED STATES PATENT

INVENTORS: Donald C. Englin
Kelvin S. Vartti
James L. Federici

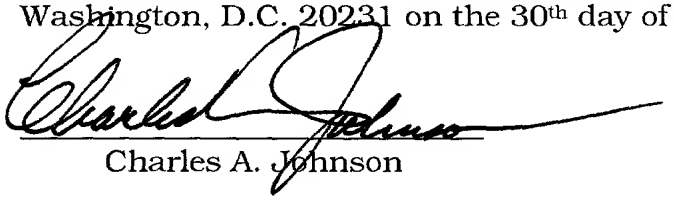
INVENTION: METHOD FOR MANAGING FLUSHES WITH THE CACHE

DOCKET NUMBER: RA 5265 (33012/294/101))

Unisys Corporation
Charles A. Johnson
P O Box 64942 - MS 4773
St. Paul, MN 55164
Attorney for Applicant
Reg. No.: 20,852

SPECIFICATION

CERTIFICATE UNDER 37 CFR 1.10: The undersigned hereby certifies that this transmittal letter and the paper of papers, as described hereinabove, are being deposited in the United States Postal Service, "Express Mail Post Office to Addressee" having an Express Mail mailing label number of EL 027383825 US, in an envelope addressed to: ASSISTANT COMMISSIONER OF PATENTS, Washington, D.C. 20231 on the 30th day of August, 2000.



Charles A. Johnson

METHOD FOR MANAGING FLUSHES WITH THE CACHE

CROSS REFERENCE TO CO-PENDING APPLICATIONS

The present application is related to co-pending U.S. Patent Application Serial No. _____, filed _____, entitled Cooperative Hardware and Microcode Control System for Pipelined Instruction Execution; U.S. Patent Application Serial No. _____, filed _____, entitled Method for Improved First Level Cache Coherency; U.S. Patent Application Serial No. _____, filed _____, entitled Method for Avoiding Delays During SNOOP Requests; U.S. Patent Application Serial No. _____, filed _____, entitled Leaky Cache Mechanism; and U.S. Patent Application Serial No. _____, filed _____, entitled Data Coherency Protocol for Multi-level Cached High Performance Multiprocessor System, assigned to the assignee of the present invention and incorporated herein by reference.

BACKGROUND OF THE INVENTION

1. **Field of the Invention:** - The present invention relates generally to data processing systems employing multiple instruction processors and more particularly relates to multiprocessor data processing systems employing multiple levels of cache memory.

2. **Description of the Prior Art:** - It is known in the art that the use of multiple instruction processors operating out of common memory can produce problems associated with the processing of obsolete memory data by a first processor after that memory data has been updated by a second processor. The first attempts at solving this problem tended to use logic to lock processors out of memory spaces being updated. Though this is appropriate for rudimentary applications, as systems become more complex, the additional hardware and/or operating time required for the setting and releasing of locks can not be justified, except for security purposes. Furthermore, reliance on such locks directly prohibits certain types of applications such as parallel processing.

The use of hierarchical memory systems tends to further compound the problem of data obsolescence. U.S. Patent No. 4,056,844 issued to Izumi shows a rather early approach to a solution. The system of Izumi utilizes a buffer memory dedicated to each of the processors in the system. Each processor accesses a buffer address array to determine if a particular data element is present in its buffer memory. An additional bit is added to the buffer address array to indicate invalidity of the corresponding data stored in the buffer memory. A set invalidity bit indicates that the main storage has been altered at that location since loading of the buffer memory. The validity bits are set in accordance with the memory store cycle of each processor.

U.S. Patent No. 4,349,871 issued to Lary describes a bussed architecture having multiple processing elements, each having a dedicated cache memory. According to the Lary design, each processing unit manages its own cache by monitoring the memory bus. Any invalidation of locally stored data is tagged to prevent use of obsolete data. The overhead associated with this approach is partially mitigated by the use of special purpose hardware and through interleaving the validity determination with memory accesses within the pipeline. Interleaving of invalidity determination is also employed in U.S. Patent No. 4,525,777 issued to Webster et al.

Similar bussed approaches are shown in U.S. Patent No. 4,843,542 issued to Dashiell et al, and in U.S. Patent No. 4,755,930 issued to Wilson, Jr. et al. In employing each of these techniques, the individual processor has primary responsibility for monitoring the memory bus to maintain currency of its own cache data. U.S. Patent No. 4,860,192 issued to Sachs et al, also employs a bussed architecture but partitions the local cache memory into instruction and operand modules.

U.S. Patent No. 5,025,365 issued to Mathur et al, provides a much enhanced architecture for the basic bussed approach. In Mathur et al, as with the other bussed systems, each processing element has a dedicated cache resource. Similarly, the cache resource is responsible for monitoring the system bus for any collateral memory accesses which would invalidate local data.

Mathur et al, provide a special snooping protocol which improves system throughput by updating local directories at times not necessarily coincident with cache accesses. Coherency is assured by the timing and protocol of the bus in conjunction with timing of the operation of the processing element.

An approach to the design of an integrated cache chip is shown in U.S. Patent No. 5,025,366 issued to Baror. This device provides the cache memory and the control circuitry in a single package. The technique lends itself primarily to bussed architectures. U.S. Patent No. 4,794,521 issued to Ziegler et al, shows a similar approach on a larger scale. The Ziegler et al, design permits an individual cache to interleave requests from multiple processors. This design resolves the data obsolescence issue by not dedicating cache memory to individual processors. Unfortunately, this provides a performance penalty in many applications because it tends to produce queuing of requests at a given cache module.

The use of a hierarchical memory system in a multiprocessor environment is also shown in U.S. Patent No. 4,442,487 issued to Fletcher et al. In this approach, each processor has dedicated and shared caches at both the L1 or level closest to the processor and at the L2 or intermediate level. Memory is managed by permitting more than one processor to operate upon a single data block only when that data block is placed in shared cache. Data blocks in dedicated or private cache are essentially locked out until placed within a shared memory element. System level memory management is

accomplished by a storage control element through which all requests to shared main memory (i.e. L3 level) are routed. An apparent improvement to this approach is shown in U.S. Patent No. 4,807,110 issued to Pomerene et al. This improvement provides prefetching of data through the use of a shadow directory.

A further improvement to Fletcher et al, is seen in U.S. Patent No. 5,023,776 issued to Gregor. In this system, performance can be enhanced through the use of store around L1 caches used along with special write buffers at the L2 intermediate level. This approach appears to require substantial additional hardware and entails yet more functions for the system storage controller.

Inherent in architectures which employ cache memory, is that the storage capacity is substantially less than the memory located at lower levels in the hierarchy. As a result, memory locations within the cache memory must often be cleared for use by other data quantities more recently needed by the instruction processor. For store-in cache memories, this means that those quantities modified by the instruction processor must first be rewritten to system memory before the corresponding location is available to store newly requested data. This "flushing" process tends to delay the availability of the newly requested data.

SUMMARY OF THE INVENTION

The present invention overcomes the problems found in the prior art by providing a method of and apparatus for improving the efficiency of cache memory within a system. This enhancement to efficiency is accomplished through a novel technique for managing the flushing process.

The preferred mode of the present invention includes up to four main memory storage units. Each is coupled directly to each of up to four "pod"s. Each pod contains a level three cache memory coupled to each of the main memory storage units. Each pod may also accommodate up to two input/output modules.

Each pod may contain up to two sub-pods, wherein each sub-pod may contain up to two instruction processors. Each instruction processor has two separate level one cache memories (one for instructions and one for operands) coupled through a dedicated system controller, having a second level cache memory, to the level three cache memory of the pod.

Unlike many prior art systems, both level one and level two cache memories are dedicated to an instruction processor within the preferred mode of the present invention. The level one cache memories are of two types. Each instruction processor has an instruction cache memory and an operand cache memory. The instruction cache memory is a read-only cache memory primarily

having sequential access. The level one operand cache memory has read/write capability. In the read mode, it functions much as the level one instruction cache memory. In the write mode, it is a semi-store-in cache memory, because the level two cache memory is also dedicated to the instruction processor.

In accordance with the present invention, the level two cache memory is of the store-in type. Therefore, the most current value of an operand which is modified by the corresponding instruction processor is first located within the level two cache memory. When the replacement algorithm for the level two cache memory determines that the location of that operand must be made available for newly requested data, that operand must be "flushed" into the lower level memory to avoid a loss of the most current value.

Waiting for flushing of the old data before requesting the new data induces unacceptable latency. Therefore, according to the present invention, a flush buffer is provided for temporary storage of the old data during the flushing process. Though this temporary storage appears at first to be a mere extension to the level two storage capacity, it greatly enhances efficiency because the flush process really does not need to utilize the level two cache memory.

The old data is moved from the level two cache memory to the flush buffer as soon as the replacement algorithm has determined which data to move, and the newly requested data is requested from the lower level memory. The flush process subsequently occurs from the flush buffer to the lower level of memory without further

reference to the level two cache. Furthermore, locations within the level two cache memory are made available for the newly requested data well before that data has been made available from the lower level memory.

BRIEF DESCRIPTION OF THE DRAWINGS

Other objects of the present invention and many of the attendant advantages of the present invention will be readily appreciated as the same becomes better understood by reference to the following detailed description when considered in connection with the accompanying drawings, in which like reference numerals designate like parts throughout the figures thereof and wherein:

FIG. 1 is an overall block diagram of a fully populated system in accordance with the present invention;

FIG. 2 is a schematic block diagram of one pod;

FIG. 3 is a schematic block diagram of one instruction processor along with its dedicated system controller;

FIG. 4 is a detailed diagram of the flush process of the present invention; and

FIG. 5 is a detailed diagram showing the flush buffers of the present invention.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

FIG. 1 is an overall block diagram of fully populated data processing system according to the preferred mode of the present invention. This corresponds to the architecture of a commercial system of Unisys Corporation termed "Voyager".

The main memory of the system consists of up to four memory storage units, MSU 10, MSU 12, MSU 14, and MSU 16. Being fully modular, each of these four memory storage units is "stand-alone" and independent of one another. Each has a separate point-to-point dedicated bi-directional interface with up to four "pods", POD 18, POD 20, POD 22, POD 24. Again, each of the up to four pods is separate and independent of one another.

The contents of POD 20 are shown by way of example. For the fully populated system, POD 18, POD 22, and POD 24 are identical to POD 20. The interface between POD 20 and each of the four memory storage units (i.e., MSU 10, MSU 12, MSU 14, and MSU 16), is via a third level cache memory designated cached interface, CI 26, in this view. CI 26 couples with two input/output controllers, I/O Module 44 and I/O Module 46, and two sub-pods, SUB 28 and SUB 30. A more detailed explanation of the POD 20 is provided below.

The above described components are the major data handling elements of the system. In the fully populated system shown, there

are sufficient components of each type, such that no single hardware failure will render the complete system inoperative. The software employed within the preferred mode of the present system utilizes these multiple components to provide enhanced reliability for long term operation.

The remaining system components are utilitarian rather than data handling. System Oscillator 32 is the primary system time and clocking standard. Management System 34 controls system testing, maintenance, and configuration. Power Controller 36 provides the required electrical power. System Oscillator 38, Management System 40, and Power Controller 42 provide completely redundant backup capability.

FIG. 2 is a more detailed block diagram of POD 20. The level three cache memory interfaces directly with the memory storage units via TLC Controller 26 (see also Fig. 1). The actual storage for the level three cache memory is TLC SRAMS 48. As indicated this static random access memory consists of eight 16 byte memory chips.

Subpod 28 and subpod 30 each contain up to two individual instruction processors. These are designated Voyager IP 50, Voyager IP 52, Voyager IP 54, and Voyager IP 56. As explained in detail below, each contains its own system controller. In accordance with the preferred mode of the present invention, these instruction processors need not all contain an identical software architecture.

FIG. 3 is a more detailed block diagram of Voyager IP 50, located within Subpod 28, located within POD 20 (see also Figs. 1 and 2). As explained above, each instruction processor has a dedicated system controller having a dedicated level two cache memory. Instruction processor 64 has two dedicated level one cache memories (not shown in this view). One level one cache memory is a read-only memory for program instruction storage. Instruction processor 64 executes its instructions from this level one cache memory. The other level one cache memory (also not shown in this view) is a read/write memory for operand storage.

Instruction processor 64 is coupled via its two level one cache memories and dedicated system controller 58 to the remainder of the system. System controller 58 contains input logic 74 to interface with instruction processor 64. In addition, data path logic 70 controls movement of the data through system controller 58. The utilitarian functions are provided by Locks, Dayclocks, and UPI 62.

The remaining elements of system controller 58 provide the level two cache memory functions. SLC data ram 66 is the data actual storage facility. Control logic 70 provides the cache management function. SLC tags 72 are the tags associated with the level two cache memory. FLC-IC Dup. Tags 76 provides the duplicate tags for the level one instruction cache memory of instruction

processor 64. Similarly, FLC-OC Dup. Tags 78 provides the duplicate tags for the level one operand cache memory of instruction processor 64. For a more complete discusses of this duplicate tag approach, reference may be made with the above identified co-pending and incorporated U.S. Patent Applications.

FIG. 4 is a detailed functional diagram showing the flushing process of the preferred mode of the present invention. Following a level one cache memory miss, a data request is made from level one operand cache memory 114 of instruction processor 110 (see also Fig. 3). In accordance with the present invention, the data request is made on memory bus 118.

If the requested data is found within second level cache memory 122 (i.e., a cache hit), the data access occurs. However, if a cache miss occurs within second level cache memory 122 (i.e., the data is not present), a level three cache memory request is made via path 178 and memory bus 130. As soon as the data is available, it is transferred from memory bus 130 via path 180.

To provide a place to store the newly requested data, cache memory 122 may need to flush some older data, if all locations are full. The selection of which location(s) to flush is in accordance with a least recently used algorithm as modified in accordance with the above identified and incorporated co-pending patent applications. The data to be flushed is transferred to flush buffer 186 from which the data is rewritten to level three memory via bus 130. Because this data is flushed from level two cache memory 122 to flush buffer 186 before the rewrite can be accomplished, space becomes quickly available within level two cache memory 122 for accommodating the newly requested data as soon as available.

FIG. 5 is detailed diagram showing the data flow in accordance with the present invention. Upon being notified of a level two cache miss, priority logic 188 determines which locations are to be flushed. This selection is made in the manner discussed above. The location(s) to be flushed is communicated to tag RAM 190 and data RAM 196 via addressing path 192.

Access of tag RAM 190 provides a determination whether there has been any modification to the data within level two cache memory. If there has been no modification as noted within tag RAM 190, no further write operation to level three memory is required. If the data has been modified, however, path 194 notifies priority logic 188 that the modified data to be flushed must be rewritten to level three memory.

Assuming that a rewrite is necessary, the data is accessed from data RAM 196 and transferred via path 200 to temp register 198. Further latency is reduced by employing two flush buffers (i.e., flush buffer0 132 and flush buffer1 134) as shown. Temp register 198 routes the data to be rewritten to either flush buffer0 132 via path 202 or to flush buffer1 134 as each becomes available.

The data to be flushed is stored within the selected flush buffer while the rewriting process is accomplished. The data to transferred to level three memory via path 136 and bus 130.

Having thus described the preferred embodiments in sufficient detail for those of skill in the art to make and use the present invention, those of skill in the art will be readily able to apply the teachings found herein to yet other embodiments within the scope of the claims hereto attached.

WE CLAIM:

1. In a data processing system having a processor responsively coupled to a store-in cache memory which is responsively coupled to a lower level memory, the improvement comprising:

a. A flush buffer responsively coupled to said store-in cache memory and said lower level memory.

2. A data processing system according to claim 1 further comprising a tag memory responsive coupled to said store-in cache memory which indicates whether a particular location within said store-in memory has been modified by said processor.

3. A data processing system according to claim 2 further comprising a logic circuit which loads said flush buffer with data from said particular location within said store-in cache memory in response to said indication that said particular location within said store-in memory has been modified by said processor.

4. A data processing system according to claim 3 wherein said flush buffer further comprises a first flush buffer store and a second flush buffer store.

5. A data processing system according to claim 4 further

comprising a temporary register responsively coupled to said store-in cache memory, said first flush buffer store, and said second flush buffer store which routes said data from said particular location to an available one of said first flush buffer store and said second flush buffer store.

6. A data processing system comprising:

a. A processor;

b. A store-in cache memory responsively coupled to said processor;

c. A lower level memory responsively coupled to said store-in cache memory; and

d. A flush buffer responsively coupled to said store-in cache memory and said lower level memory.

7. A data processing system according to claim 6 wherein said flush buffer further comprises a first flush buffer store and a second flush buffer store.

8. A data processing system according to claim 7 further comprising:

a. A temporary register responsively coupled to said store-in cache memory, said first flush buffer store, and said second flush buffer store.

9. A data processing system according to claim 8 further comprising:

a. A tag memory responsively coupled to said store-in cache memory for indicating whether a particular location has been modified by said processor.

10. A data processing system according to claim 6 further comprising:

a. A logic circuit responsively coupled to said tag memory, said store-in cache memory, and said temporary register which routes data from said particular location from said store-in cache memory to said temporary register when said indication is that said particular location has been modified by said processor.

11. A method of flushing a store-in cache memory comprising:

a. Receiving a data request at said store-in cache memory;

b. Searching said store-in cache memory in response to said data request;

c. Experiencing a cache miss in response to said searching step;

d. Selecting a particular location within said store-in cache memory to be flushed; and

e. Transferring data from said particular location to a flush buffer.

12. A method according to claim 11 further comprising:

a. Determining whether data within said particular location was modified by a processor.

13. A method according to claim 12 further comprising:

a. Inhibiting said transferring step if said determining step determines that said data within said particular location was not modified by said processor.

14. A method according to claim 13 wherein said transferring step further comprises routing said data to the available one of a first flush buffer store and a second flush buffer store.

15. A method according to claim 14 further comprising:

a. Rewriting said data to a lower level memory following said transferring step.

16. An apparatus comprising:

a. Means for executing program instructions;

b. Means responsively coupled to said executing means for caching data on a store-in basis; and

c. Means responsively coupled to said caching means for buffering data from said caching means to be flushed.

17. An apparatus according to claim 16 further comprising:

a. Means responsively coupled to said caching means for selecting said data to be flushed.

18. An apparatus according to claim 17 wherein said buffering means further comprises a first means for storing and a second means for storing.

19. An apparatus according to claim 18 further comprising:

a. Means responsively coupled to said first storing means and said second storing means for routing said data to the available one of said first storing means and said second storing means.

20. An apparatus according to claim 19 further comprising:

a. Means responsively coupled to said caching means for determining whether said data has been modified by said executing means; and

b. Means responsively coupled to said determining means and said buffering means for inhibiting transfer of data from said caching means to said buffering means if said determining means determines that said data has not been modified by said executing means.

METHOD FOR MANAGING FLUSHES WITH THE CACHE

ABSTRACT OF THE DISCLOSURE

A method of and apparatus for improving the efficiency of a data processing system employing a multiple level cache memory system. The efficiencies result from managing the process of flushing old data from the second level cache memory. In the present invention, the second level cache memory is a store-in memory. Therefore, when data is to be deleted from the second level cache memory, a determination is made whether the data has been modified by the processor. If the data has been modified, the data must be rewritten to lower level memory. To free the second level cache memory for storage of the newly requested data, the data to be flush is loaded into a flush buffer for storage during the rewriting process.

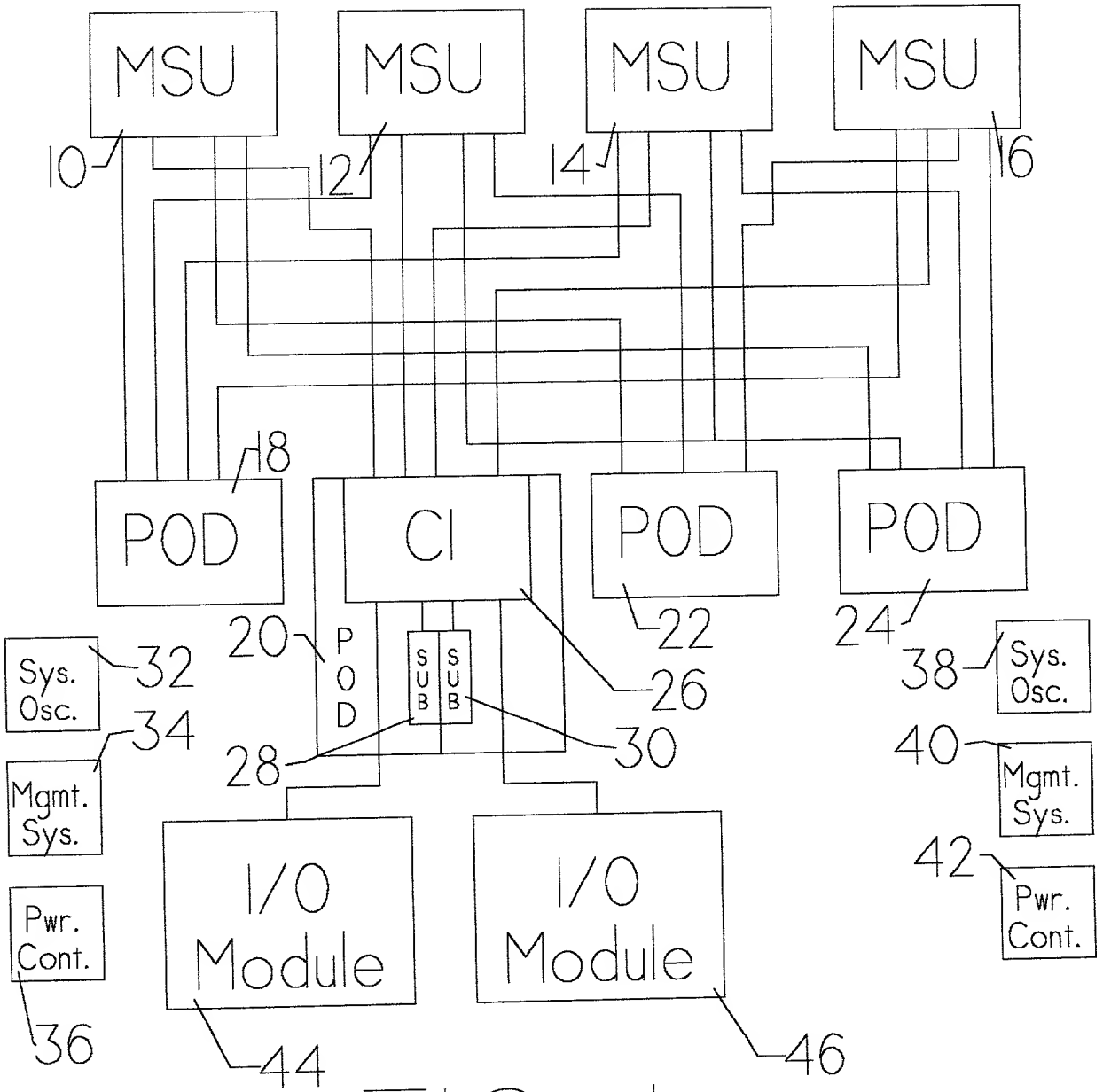

MSU
MSU
MSU
MSU
10
12
14
16
18
POD
CI
POD
POD
20
22
24
POD
SUB
SUB
26
28
30
Sys. Osc.
32
34
Mgmt. Sys.
Pwr. Cont.
36
38
Sys. Osc.
40
Mgmt. Sys.
42
Pwr. Cont.
I/O Module
I/O Module
44
46

FIG. 1

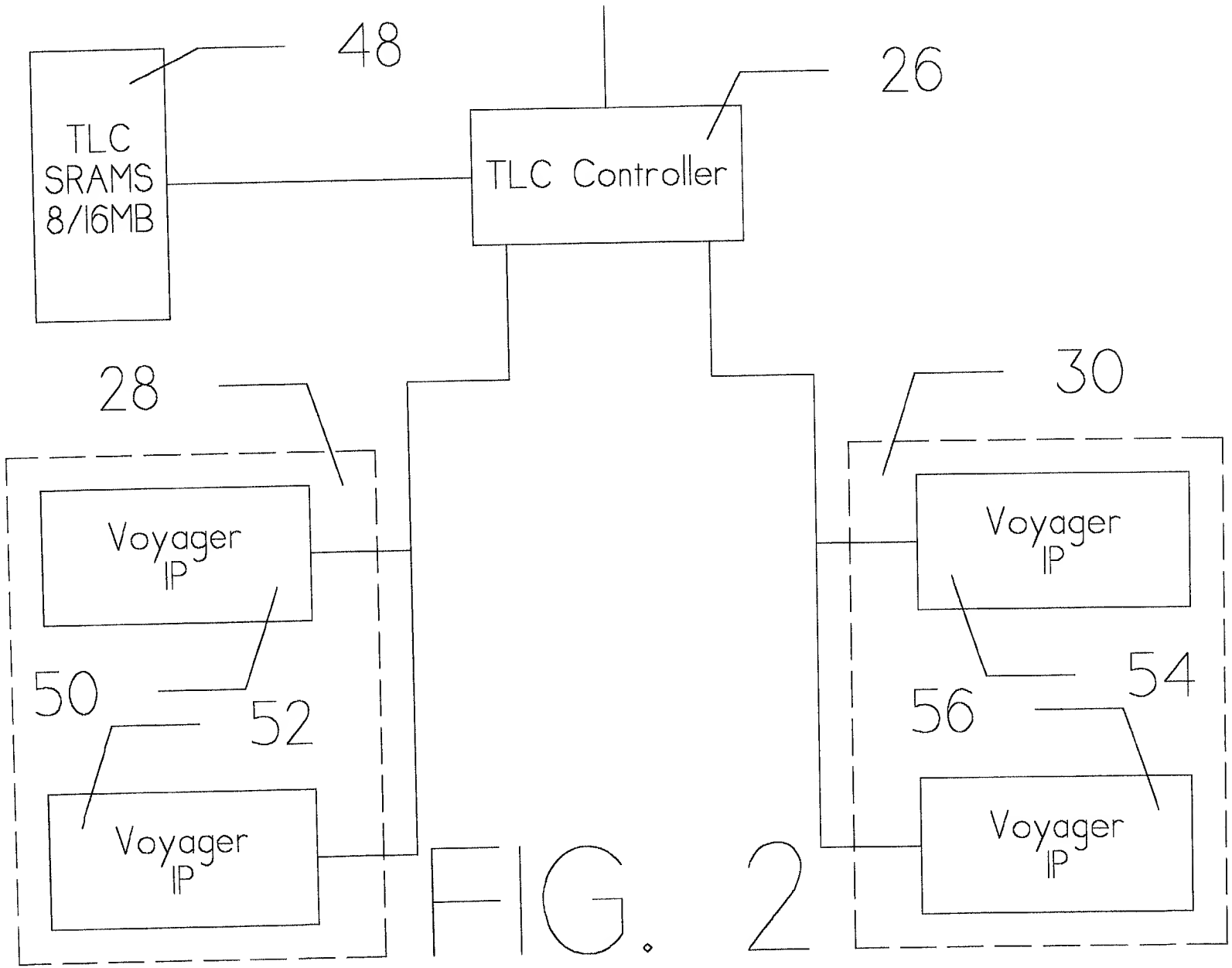

48
TLC
SRAMS
8/16MB
26
TLC Controller
28
30
Voyager
IP
Voyager
IP
50
52
54
56
Voyager
IP
Voyager
IP

FIG. 2

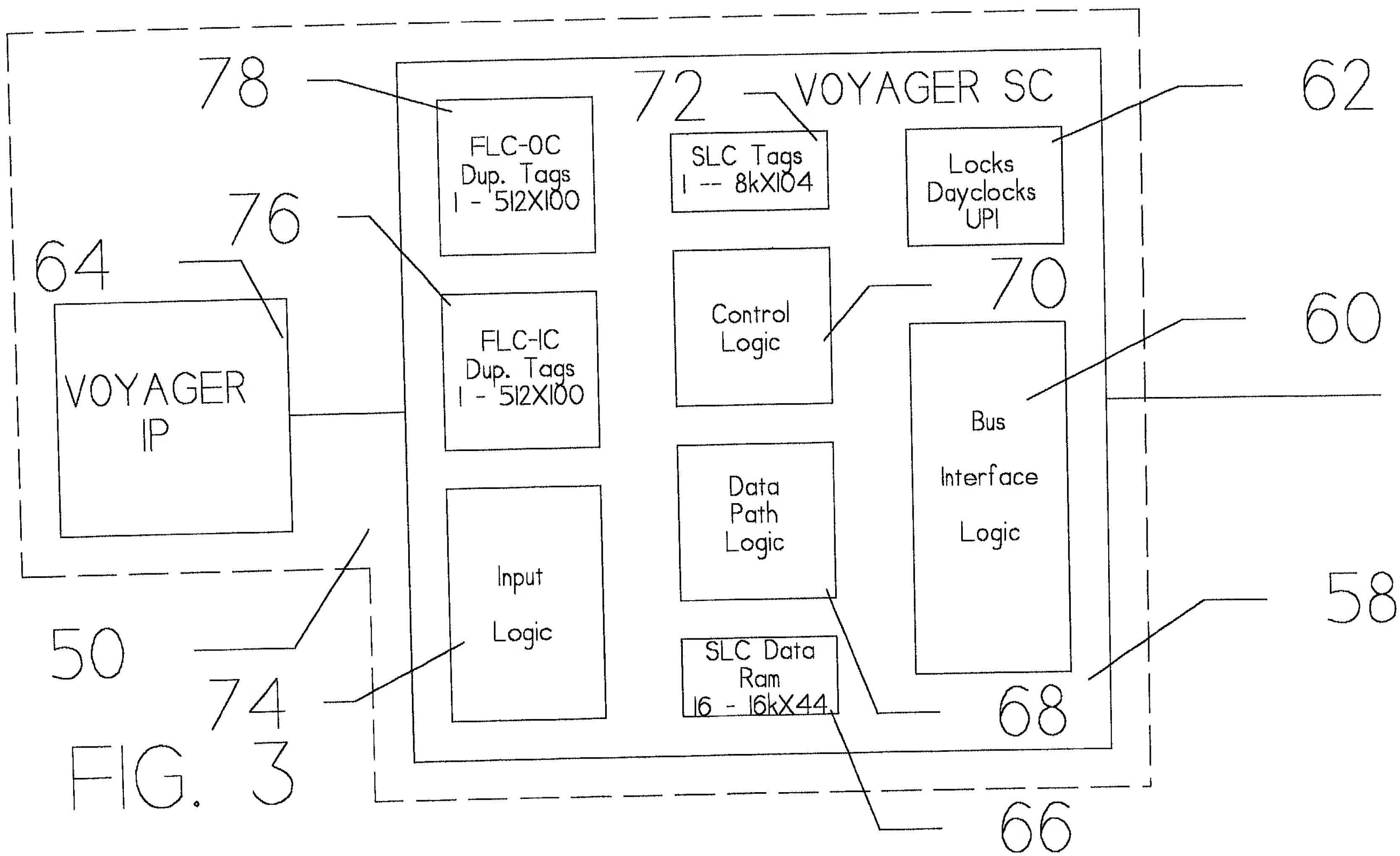

78
72
VOYAGER SC
62
FLC-0C
Dup. Tags
1 - 512X100
SLC Tags
1 -- 8kX104
Locks
Dayclocks
UPI
76
64
70
60
VOYAGER
IP
FLC-1C
Dup. Tags
1 - 512X100
Control
Logic
Bus
Interface
Logic
Data
Path
Logic
Input
Logic
58
50
SLC Data
Ram
16 - 16kX44
74
68
66

FIG. 3

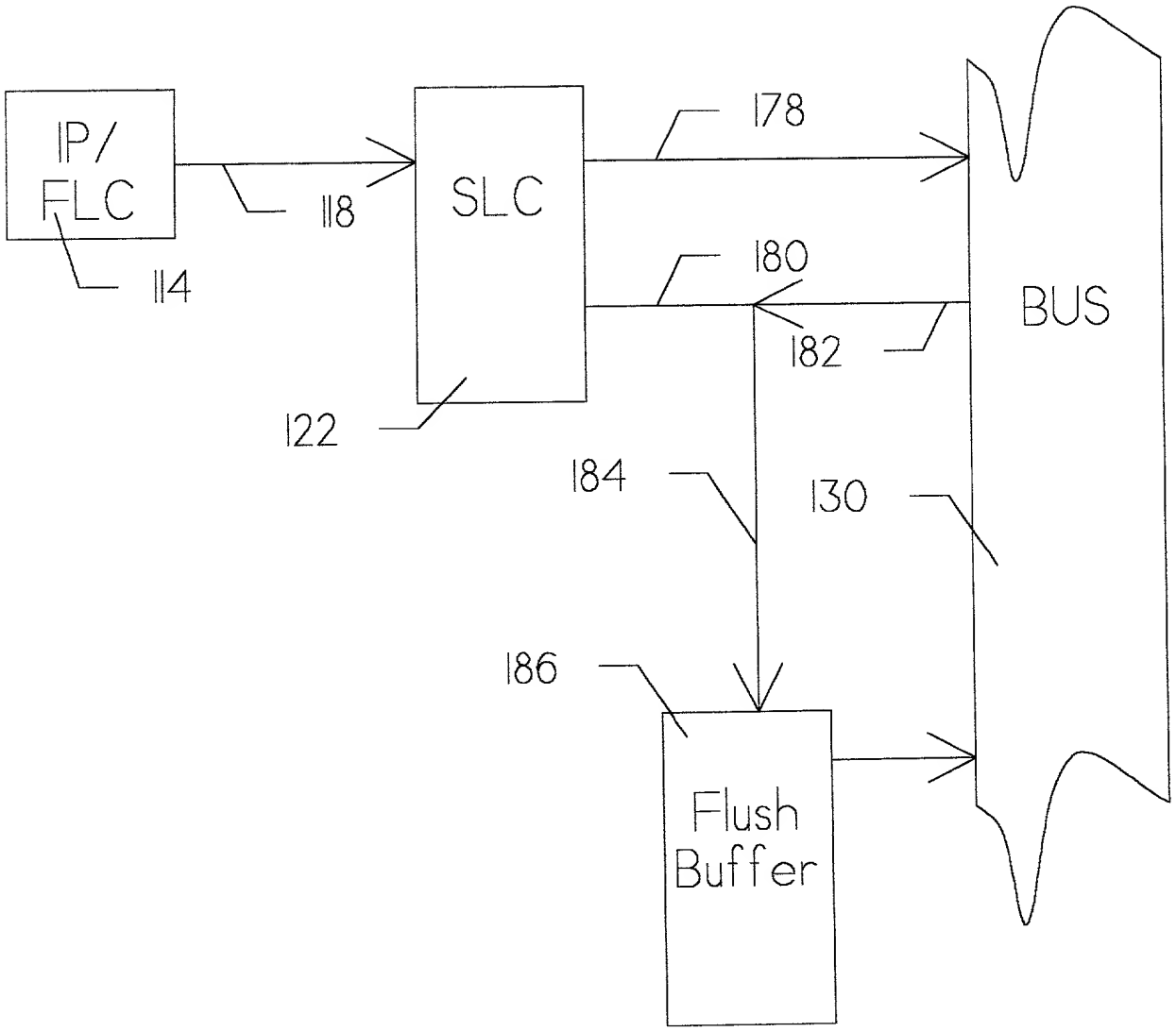


FIG. 4

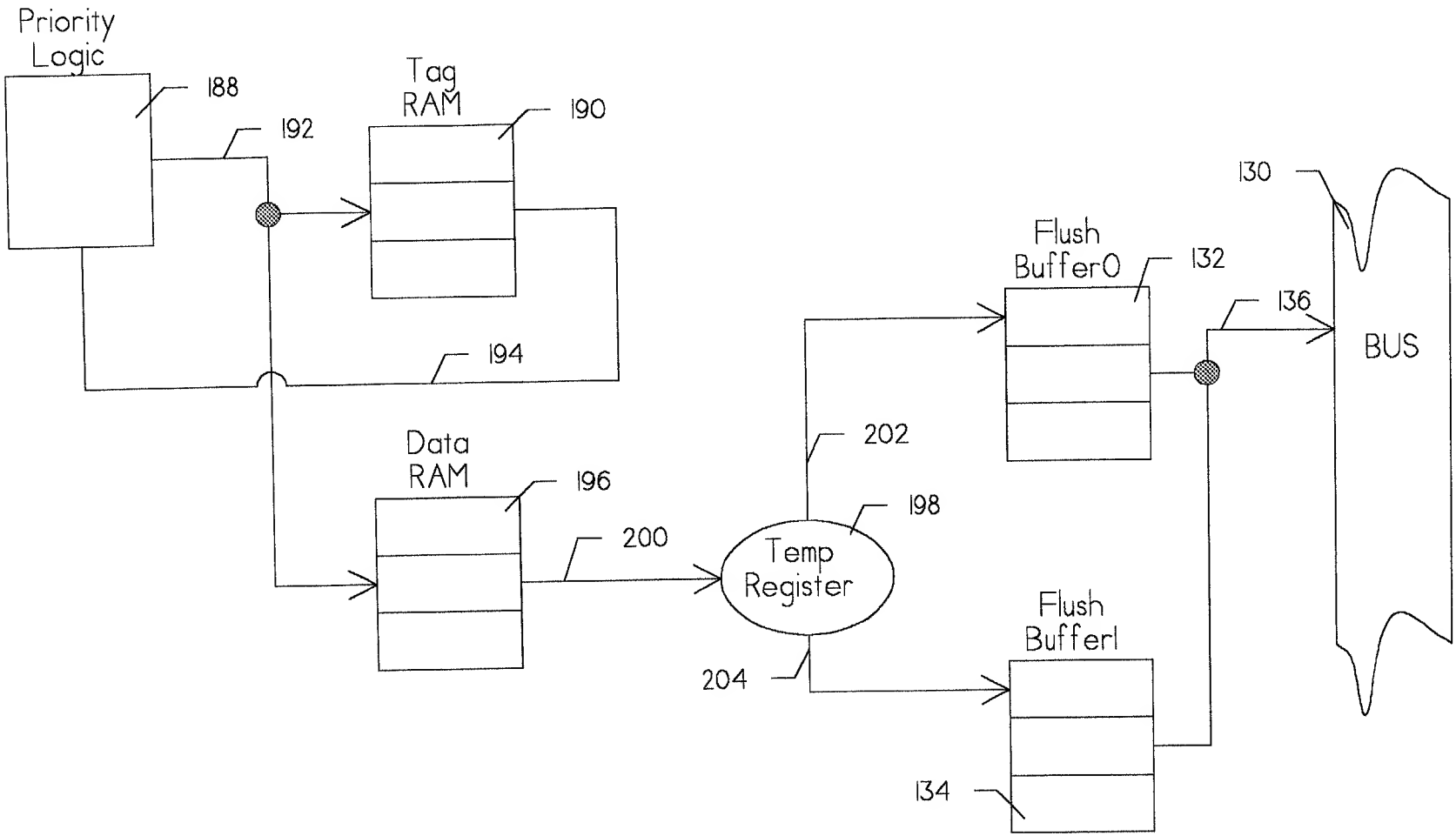

Priority
Logic
188
192
Tag
RAM
190
194
Data
RAM
196
200
Temp
Register
198
202
204
Flush
Buffer0
132
Flush
Buffer1
134
130
136
BUS

FIG. 5

COMBINED DECLARATION/POWER OF ATTORNEY FOR PATENT APPLICATION

As a below named inventor, I hereby declare that:

My residence, post office address and citizenship are as stated below next to my name.

I believe that I am the original, first and sole inventor (if only one name is listed below) or an original, first and joint inventor (if plural names are listed below) of the subject matter which is claimed and for which a patent is sought on the invention entitled METHOD FOR MANAGING FLUSHES WITH THE CACHE, the specification of which (check one)

XX is attached hereto

__ was filed on _____
as U.S. Application
Serial No. _____

__ and was amended on (if
applicable) _____

I hereby state that I have reviewed and understand the contents of the above-identified specification, including the claims, as amended by any amendment referred to above.

I acknowledge the duty to disclose information which is material to the examination of this application in accordance with Title 37, Code of Federal Regulations, §1.56(a).

I hereby claim foreign priority benefit(s) under Title 35, United States Code §119 of any foreign application(s) for patent or inventor's certificate listed below and have also identified below any foreign application(s) for patent or inventor's certificate having a filing date before that of the application on which priority is claimed:

Prior Foreign Application(s)			Priority Claimed	
(Number)	(Country)	(Day/Month/Year Filed)	YES	NO
(Number)	(Country)	(Day/Month/Year Filed)	YES	NO
(Number)	(Country)	(Day/Month/Year Filed)	YES	NO

I hereby claim the benefit under Title 35, United States Code, §120 of any United States application(s) listed below and, insofar as the subject matter of each of the claims of this application is not disclosed in the prior United States application in the manner

provided by the first paragraph of Title 35, United States Code, §112, I acknowledge the duty to disclose material information as defined in Title 37, Code of Federal Regulations, §1.56(a) which occurred between the filing date of the prior application and the national or PCT international filing date of this application:

(Serial No.)	(Filing Date)	(Status) (patented, pending, abandoned)
(Serial No.)	(Filing Date)	(Status-patented, pending, abandoned)

POWER OF ATTORNEY: As a named inventor, I hereby appoint the following attorney(s) and/or agent(s) to prosecute this application and transact all business in the Patent and Trademark Office connected therewith.

John L. Rooney, Reg. No. 28,898;
Lawrence M. Nawrocki, Reg. No. 29,333;
Wayne A. Sivertson, Reg. No. 25,645;
Richard C. Stempkovski, Jr., Reg. No. 45,130;
Jeffery L. Cameron, Reg. No. 43,527;
Donald A. Jacobson, Reg. No. 22,308; and
Charles A. Johnson, Reg. No. 20,852

Send correspondence to:

Charles A. Johnson
Unisys Corporation
Law Department
M.S. 4773
2470 Highcrest Road
Roseville, Minnesota 55113

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or any patent issued thereon, I further declare that I understand the content of this declaration.

Full name of sole or first inventor Donald C. Englin
Inventor's Signature Donald C Englin Date 8/23/2000
Residence 436 Dudley Avenue
Shoreview, Minnesota 55126 Citizenship U.S.A.
Post Office Address 436 Dudley Avenue
Shoreview, Minnesota 55126

Full name of second or joint inventor Kelvin S. Vartti
Inventor's Signature Kelvin S. Vartti Date 8-23-2000
Residence 11893 Imperial Avenue North
Hugo, Minnesota 55038 Citizenship U.S.A.
Post Office Address 11893 Imperial Avenue North
Hugo, Minnesota 55038

Full name of third or joint inventor James L. Federici
Inventor's Signature James L Federici Date 8/29/00
Residence 6706 Partridge Place
Lino Lakes, Minnesota 55014 Citizenship U.S.A.
Post Office Address 6706 Partridge Place
Lino Lakes, Minnesota 55014

1.56 Duty to disclose information material to patentability.

(a) A patent by its very nature is affected with a public interest. The public interest is best served, and the most effective patent examination occurs when, at the time an application is being examined, the Office is aware of and evaluates the teachings of all information material to patentability. Each individual associated with the filing and prosecution of a patent application has a duty of candor and good faith in dealing with the Office, which includes a duty to disclose to the Office all information known to that individual to be material to patentability as defined in this section. The duty to disclose information exists with respect to each pending claim until the claim is cancelled or withdrawn from consideration, or the application becomes abandoned. Information material to the patentability of a claim that is cancelled or withdrawn from consideration need not be submitted if the information is not material to the patentability of any claim remaining under consideration in the application. There is no duty to submit information which is not material to the patentability of any existing claim. The duty to disclose all information known to be material to patentability is deemed to be satisfied if all information known to be material to patentability of any claim issued in a patent was cited by the Office or submitted to the Office in the manner prescribed by §§1.97(b)-(d) and 1.98. However, no patent will be granted on an application in connection with which fraud on the Office was practiced or attempted or the duty of disclosure was violated through bad faith or intentional misconduct. The Office encourages applicants to carefully examine:

(1) prior art cited in search reports of a foreign patent office in a counterpart application, and

(2) the closest information over which individuals associated with the filing or prosecution of a patent application believe any pending claim patentably defines, to make sure that any material information contained therein is disclosed to the Office.

(b) Under this section, information is material to patentability when it is not cumulative to information already of record or being made of record in the application, and

(1) It establishes, by itself or in combination with other information, a prima facie case of unpatentability of a claim; or

(2) It refutes, or is inconsistent with, a position the applicant takes in:

(i) Opposing an argument of unpatentability relied on by the Office, or
(ii) Asserting an argument of patentability.

A prima facie case of unpatentability is established when the information compels a conclusion that a claim is unpatentable under the preponderance of evidence, burden-of-proof standard, giving each term in the claim its broadest reasonable construction consistent with the specification, and before any consideration is given to evidence which may be submitted in an attempt to establish a contrary conclusion of patentability.

(c) Individuals associated with the filing or prosecution of a patent application within the meaning of this section are:

(1) Each inventor named in the application:

(2) Each attorney or agent who prepares or prosecutes the application; and

(3) Every other person who is substantively involved in the preparation or prosecution of the application and who is associated with the inventor, with the assignee or with anyone to whom there is an obligation to assign the application.

(d) Individuals other than the attorney, agent or inventor may comply with this section by disclosing information to the attorney, agent, or inventor.